Software Development Products

Software Defect Update

Intel® Fortran Compiler for Linux* and Windows*

19th January 2005

Legal Disclaimer

This document is a compilation of software and software documentation defects, and software specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems or tools.

Except as expressly provided in Intel's standard terms and conditions of sale for the Intel software product or in the Intel software license agreement accompanying the Intel software product, the Intel software product is provided "as is", without warranty of any kind, whether express, implied or statutory, including but not limited to a warranty of merchantability, noninfringement of intellectual property, or fitness for any particular purpose.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY EXPRESS, IMPLIED, OR STATUTORY WARRANTY OF ANY KIND INCLUDING BUT NOT LIMITED TO WARRANTIES OF MECHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy, completeness or utility of any information contained herein. Intel may make changes to these materials, or to the Intel products described therein, at any time without notice. Intel makes no commitment to update these materials.

Independent companies manufacture the third–party products that are mentioned in this document. Intel is not responsible for the quality of third–party products and makes no representation or warranty regarding such products. The third–party supplier remains solely responsible for the design, manufacture, sale and functionality of its products.

If you have any questions about the issues discussed in this report, please open a support issue at https://premier.intel.com/

Linux*

22214	Fortran compiler treats 'if_' as a keyword rather than a label.
21708	Documentation files installed with execute permissions enabled
14358	Compiler does not support vectorization of complex data types
24814	LDB cannot distinguish source files with the same name
26080	Erroneous re−type warning for Cray−style pointer dummy argument
26550	Flush of unwritten file causes 'FLUSH FAILED'
30073	Idb cannot resolve local variable
26758	minval/maxval may produce incorrect results when a pointer is used as an argument
24114	The maximum array size is limited to (2**31−1) bytes on a IA32 machine
29260	Compilation hangs at all optimization levels when building the POP software package
30098	ieee_flags returns blanks in "out" parameter
31310	Using idb under GNU Emacs v21 fails with error
31200	OpenMP code using many OMP SECTIONS causes long compile times
31098	MATMUL matrix multiply intrinsic function produces incorrect results
31677	Using −prof_use −O3 switches may produce incorrect program output
31945	Problem with ddd and idb on SuSE* 8.1
32143	Loop is not auto−parallelized at −O3
32108	Install script on Itanium(R) systems may report error message about missing FLEXlm* license
32240	PRIVATE attribute in a derived data type may produce incorrect results
32306	Jump into DO loop
32393	Using −O3 with the matmul function may produce incorrect results
33555	Tabs in Fortran source may cause a syntax error
33731	Calls to FSTAT do not compile as documented
43808	cachesize function returns incorrect values

34961	Zero length .o file created upon abnormal compiler termination
37124	Array Section Argument Corrupted at Run Time
38019	Internal compiler error when using the –openmp_report2 compiler switch
39520	PRESENT intrinsic may return incorrect value for use inside an internal subroutine
41381	Scalar array assignment doesn't work in nested internal subroutine
37717	INTEL (flexlm) vendor daemon fails to start
41760	Integer calculations of the form 2**k when k is a negative number produce incorrect results
41783	fpp flags an error on HOLLERITH constants containing '!'
42931	real*4 arrays with module–variable size used in OpenMP* reductions cause Abort

Windows*

23927	Unnecessary Float–to–Double Conversions Hinder Performance
24678	EDB does not recognize executable program path names containing blank characters
26140	Character constant with KIND not accepted in format field of WRITE
29518	Apps with source files with names beginning with 'etrip' may fail at runtime
31061	***Fatal: bad switch value 0 at line XXXX in file proton/libi/f90ioa.c
31267	QWIN.HLP help file missing from the compiler package
30924	Unable to compile individual Fortran file in Microsoft* Visual Studio .NET* environment
31493	Using BSEARCHQQ Function Causes Syntax Error 394 – Misplaced Statement
31492	C preprocessor "stringization" and "merging" operators not recognized by Fortran preprocessor
31556	The "build analysis" feature is only available after the initial build of an open solution
31594	Fortran project relinks on every build
32833	EDB and Visual Studio* .NET debugger cannot show local variables in subroutines with entry points
33220	Compiler interprets \" as escape sequence in command line input
34275	The TRANSFER function produces incorrect result when used in an initialization expression
34682	Quad precision complex operands may produce incorrect results when compiled with /QxK or /QxW
36243	Modifying a single file forces rebuild of entire solution

38042	Generating debug info for two dimensional array causes internal error
39033	16−byte alignment and type declaration causes an internal compiler error
41316	Error when initializing an array using mixed−mode expression
42004	GETARG routine does not work correctly with integer*2 data type for its third argument
47792	No warning message issued when division by zero occurs

Linux*

Title	Fortran compiler treats 'if_' as a keyword rather than a label.
Reference #	22214
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0, 7.0
Operating System	Red Hat* 6.2
Problem Description	The Intel(R) Fortran Compiler for Linux* treats a label starting with if_ as a keyword rather than a label. In the following example: if_i: if(i.eq.1)then i=i+1 else if_i i=i−1 endif if_i the compiler treats the "else" as an "else if" rather than a label. It compiles if another label name is chosen.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034 or higher. You may download and install the latest product update from the Premier Support web site at . You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Documentation files installed with execute permissions enabled
Reference #	21708
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0, 7.0
Operating System	Red Hat* 7.1
Problem Description	Most of the compiler documentation is installed with execute permissions for user, group, and world.

	For example: $ ls –l /opt/intel/compiler60/docs total 12344 –rwxr–xr–x 1 root root 282345 Jan 4 2002 asm_lan.pdf –rwxr–xr–x 1 root root 274778 Feb 27 2002 asm_ug.pdf ...
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Compiler does not support vectorization of complex data types
Reference #	14358
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0, 7.0
Operating System	Red Hat* 7.1

Problem Description	The Intel Fortran Compiler will not vectorize complex data types. The following test routine fails to vectorize the complex data structure, but does vectorize if the data types are real.
	
	$ cat p.f90
	subroutine test(x,y,z)
	complex, dimension(1:100)::x,y,z
	
	x = y*z
	
	return
	end
	
	$ ifc –c –xW p.f90 –vec_report3
	external subroutine PARVEC
	p.f90(4) : (col. 0) remark: loop was not vectorized: data type unsupported on given target architecture.
	
	7 Lines Compiled

Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	LDB cannot distinguish source files with the same name
Reference #	24814
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0, 7.0
Operating System	Red Hat* 7.1
Problem Description	If an application contains two files of the same name that are in different subdirectories and built into different libraries, LDB cannot distinguish between the two files and warns of this condition as follows:: Can not distinguish between the following files: /home/test/libdir/foof.f /home/test/otherdir/foof.f
Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	Erroneous re−type warning for Cray−style pointer dummy argument
Reference #	26080
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0,7.0
Operating System	Red Hat* 7.1
Problem Description	Compilation of the following subroutine causes issuance of an erroneous warning level message regarding Cray−style pointer variable re−typing as shown below: subroutine foo (ib) implicit none real b pointer (ib, b) end Warning at compilation: pointer (ib, b) Warning 114: Pointer variable has already been declared − retyped as INTEGER
Resolution/Status	This is a known issue and may resolved in a future release. Additional information on Cray−style pointer support including type handling is available in the Intel ® Fortran Programmer's Reference manual.

Title	Flush of unwritten file causes 'FLUSH FAILED'
Reference #	26550
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0
Operating System	Red Hat* 7.1
Problem Description	Use of FLUSH on an un–written file generates an incorrect failure. Below is a simple test and output. program flushtest open(10,FILE="output1") open(11,FILE="output2") write(10,*) "Hello!" print *, "Flushing file written to" call flush(10) print *, "Flushing file not written to" call flush(11) stop end Program output: Flushing file written to Flushing file not written to FLUSH––FAILED:: Invalid argument
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_6.0.1.304 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	ldb cannot resolve local variable
Reference #	30073
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0
Operating System	Red Hat* 7.2 – delete
Problem Description	ldb may be unable to disambiguate between local variables in different contexts that share the same name. If the sample program below is compiled with ifc using the –g switch (produce symbolic debug info), ldb can resolve local

	variable 'x' from within the context of TEST, however, it cannot resolve the name 'x' from within the context of Subroutine f1. For example, the ldb print command returns the following error from within the context of f1: (ldb) print x ldb error: Name "x" is ambiguous (possibly bad debug information) PROGRAM TEST IMPLICIT NONE INTEGER :: x x=1 CALL f1(x) STOP CONTAINS SUBROUTINE f1(x) IMPLICIT NONE INTEGER :: x PRINT *,x END SUBROUTINE f1 END PROGRAM TEST
Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	minval/maxval may produce incorrect results when a pointer is used as an argument
Reference #	26758
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0,7.0
Operating System	Red Hat* 7.1
Problem Description	Compiler may produce an incorrect result of the minval (or maxval) function when a pointer is used as an argument.
Resolution/Status	This problem has been resolved in the Intel(R) Fortran Compiler 7.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	The maximum array size is limited to (2**31−1) bytes on a IA32 machine

Reference #	24114
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0,7.0
Operating System	Red Hat* 7.1
Problem Description	The maximum array size seems to be limited to (2**31−1) bytes on IA32. Is there any way to exceed this limitation? I presently get this error message: "In program unit MAIN the size of array A1 exceeds the implementation limit (2**31−1)"
Resolution/Status	This is a known issue that may be resolved in a future product release. As a workaround, if it is possible to divide the work of your application into processes, MPI can be used as each process will have its own 2 gigabyte address space.

Title	Compilation hangs at all optimization levels when building the POP software package
Reference #	29260
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.2
Problem Description	Compilation hangs at all optimization levels when compiling the vertical_mix.f module which is part of the Parallel Ocean Program (POP) software package.
Resolution/Status	This problem has been resolved in a product update with package ID l_FC_PU_7.0.013 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	ieee_flags returns blanks in "out" parameter
Reference #	30098
Product	Intel(R) Fortran Compiler for Linux*
Version	6.0,7.0
Operating System	Other (specify below)
Problem Description	The ieee_flags portability intrinsic function may return a blank value in the "out" parameter string rather than the expected string. For example,

	status=ieee_flags('get', 'exception', in, out) write(*,*) out should print out 'division' to stdout when a floating point divide by zero exception is caught. Instead, you will only see blanks.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_pu_7.0.083 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Using idb under GNU Emacs v21 fails with error
Reference #	31310
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1
Problem Description	When using the Linux Application Debugger, idb, under Emacs v21, after loading idb.el, the command M−x idb fails with the error "Can't figure out which 'gud' is being used." This is due to a change in the syntax of the function `gud−common−init' between GNU Emacs 20.7 and 21. In 20.7, gud−common−init takes 4 arguments: (gud−common−init COMMAND−LINE MASSAGE−ARGS MARKER−FILTER FIND−FILE) In Emacs 21, the last argument has been made optional: (gud−common−init COMMAND−LINE MASSAGE−ARGS MARKER−FILTER &optional FIND−FILE)
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_pc_8.0.039 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	OpenMP code using many OMP SECTIONS causes long compile times

Reference #	31200
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Debian*
Problem Description	The compiler isn't checking for OMP SECTIONS directives efficiently, so if you have many of them in your code with many sub OMP SECTION directives, the compiler can take a very long time to compile.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_pu_7.0.087 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	MATMUL matrix multiply intrinsic function produces incorrect results
Reference #	31098
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1
Problem Description	The MATMUL matrix multiply intrinsic function does not work correctly when one of the matrices is a component of a derived type. The following program displays this problem: PROGRAM matrix_test TYPE special COMPLEX :: x, y ! gives incorrect result END TYPE special TYPE(special), DIMENSION(2,2) :: a COMPLEX, DIMENSION(2,2) :: b COMPLEX :: c1=(1.0,0.0) a(1,1)%x=11.0*c1; a(1,2)%x=12.0*c1 a(2,1)%x=21.0*c1; a(2,2)%x=22.0*c1 b(1,1)=c1; b(2,2)=c1 b=MATMUL(a%x,b) PRINT *, b(1,1), b(1,2) PRINT *, b(2,1), b(2,2) STOP END PROGRAM matrix_test

	$ ifc –w badcase.f90 program MATRIX_TEST 21 Lines Compiled $ ./a.out (11.00000,0.000000E+00) (21.00000,0.000000E+00) (0.000000E+00,0.000000E+00) (0.000000E+00,0.000000E+00)
Resolution/Status	This problem has been resolved in Intel(R) Fortran Compiler with package ID l_fc_p_7.1.008 or higher. You may download and install the latest product update from the premier support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Using –prof_use –O3 switches may produce incorrect program output
Reference #	31677
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1
Problem Description	The compiler may produce incorrect code for some programs when the code is compiled with the –O3 –prof_use compiler switches. This is not a general issue and does not apply to all programs. Correct code is produced using "–O1 –prof_use", or "–O2 –prof_use", or simply the "–O3" switch without "–prof_use".
Resolution/Status	This problem has been resolved in a product update with package ID l_FC_PU_7.0.017 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Problem with ddd and idb on SuSE* 8.1

Reference #	31945
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.2 – delete
Problem Description	When using the ddd debugging GUI with the idb debugger on SuSE* 8.1, the debugger often exits with Error 127.
Resolution/Status	This is a known issue that may be resolved in a future product release. As a workaround use the idb debugger from the command–line on SuSE* 8.1.

Title	Loop is not auto–parallelized at –O3
Reference #	32143
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1

Problem Description	The compiler at –O3 may misdiagnose code that contains a loop that will explicitly execute at least one time, and therefore will not parallelize it with the following diagnostic:
	
	$ efc –O3 –parallel –par_threshold0 –par_report3 –c test.f
	
	external subroutine TEST
	procedure: test
	serial loop: line 30: not a parallel candidate due to missing zero–trip test
	
	Note that this diagnostic may be correct at times. You can verify if this diagnostic is due to this issue by compiling with –O2 instead.
	
	$ efc –O2 –parallel –par_threshold0 –par_report3 –c test.f
	
	external subroutine TEST
	procedure: test
	test.f(30) : (col. 0) remark: LOOP WAS AUTO–PARALLELIZED.
	parallel loop: line 30
	shared: {"da", "dx", "dy"}
	private: {"i"}
	first private: { }
	reductions: { }
	
	If the loop is parallelized, then the behavior at –O3 is due to this compiler issue.

Resolution/Status	This problem has been resolved in the Intel(R) Fortran Compiler 7.1 with package ID l_fc_pu_7.1.015 or higher. You may download and install the latest

	product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	Install script on Itanium(R) systems may report error message about missing FLEXlm* license
Reference #	32108
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0, 7.1
Operating System	Red Hat* 7.1
Problem Description	The install script reports the following message about missing FLEXlm* license only on an Itanium(R) system whose kernel has been compiled with the IA32 support turned off. The message is reported even though a valid FLEXlm* license may exist on the system. This problem also applies to the Intel(R) C++ Compiler for Linux* installation: $ ./install A valid FLEXlm license is required to install this product, but none were found in $INTEL_LICENSE_FILE (.:/opt/intel/licenses). Where is a valid FLEXlm license for this product? Enter directory or file, or just 'Enter' to exit.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.032 or higher. You may download and install the latest product update from the premier support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	PRIVATE attribute in a derived data type may produce incorrect results
Reference #	32240
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1
Problem Description	If a program defines a derived data type and applies the PRIVATE attribute to its components, accessing the structure components may produce incorrect results. For example, given: TYPE POSVECTOR PRIVATE ! BUG if made PRIVATE; ok if PUBLIC (default)

	REAL :: x REAL :: y REAL :: z END TYPE POSVECTOR TYPE (POSVECTOR) :: RHAT RHAT(1.0, 2.0, 3.0) REAL :: a, b, c a = RHAT%x b = RHAT%y c = RHAT%z May not assign the correct values to REAL variables a, b, and c. This problem is only known to occur on Itanium(r)–based platforms.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_bc_8.0.025 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Jump into DO loop
Reference #	32306
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	MSC.Linux

Problem Description	The compiler allows a GO TO into a DO loop, issuing only a warning instead of an error, as in the following example.
	
	j = 1
	if (j .eq. 2) go to 10
	do 20 i=1, 10
	10 a = i
	20 continue
	end
	
	$ ifc test.f90
	program TEST
	
	10 a = i
	^
	Warning 7 at (6:test.f90) : Transfer to label 10 within a DO block from (4:test.f90)

	is an extension to standard Fortran 95 10 Lines Compiled
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034_pl038 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Using –O3 with the matmul function may produce incorrect results
Reference #	32393
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0,7.1
Operating System	Red Hat* 7.1

Problem Description	Compiling the matrix multiply function matmul with –O3 may produce incorrect results The program below works correctly when compiled with –O2, but generates incorrect results when compiled with –O3 due to the implied overlap between the arrays a, b, and c in combination with inlining of the MATMUL intrinsic.
	
	$ cat test.f
	real, pointer :: a(:,:),b(:,:),c(:,:)
	real, target :: x(4,4)
	x(:,:)=1.0
	a=>x(:,:)
	b=>x(:,:)
	c=>x(:,:)
	c = matmul(a,b)
	write(*,*) c
	end
	$ efc –O3 test.f –otestO3
	main program
	
	9 Lines Compiled
	$ ./testO3
	3.000000 2.000000 5.000000 14.00000 2.000000
	22.00000 461.0000 2210.000 5.000000 461.0000
	4.517601E+10 9.205787E+16 14.00000 2210.000 9.205787E+16
	++++++++++++++

Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information,

	please visit http://www.intel.com/software/products/support.
	
	As a workaround, compile with –O2. Another workaround is to create a temporary result array, y, with no overlap (or aliasiing) with arrays a and b, with assignment of array y to array c, subsequent to the MATMUL call.
	
	$ cat wa.f
	real, pointer :: a(:,:),b(:,:),c(:,:)
	real, target :: x(4,4)
	real, target :: y(4,4)
	x(:,:)=1.0
	a=>x(:,:)
	b=>x(:,:)
	c=>y(:,:)
	c = matmul(a,b)
	x=y
	c=>x(:,:)
	write(*,*) c
	end
	$ efc –O3 wa.f –owaO3
	main program
	
	12 Lines Compiled
	$ ./waO3
	4.000000 4.000000 4.000000 4.000000 4.000000
	4.000000 4.000000 4.000000 4.000000 4.000000
	4.000000 4.000000 4.000000 4.000000 4.000000
	4.000000
	$

Title	Tabs in Fortran source may cause a syntax error
Reference #	33555
Product	Intel(R) Fortran Compiler for Linux*
Version	7.1
Operating System	Red Hat* 7.1
Problem Description	Some usage of tabs in a Fortran source file may lead to a syntax error. The sample program below contains <tab> references to represent the presence of a non–visible tab character in the source line. Usage of the leading tab character on the comment line reading 'Leading tab appears on this line' will generate a compilation syntax error for the subsequent continued Fortran source line, as shown below. $ cat tab.f <tab>program p2 <tab>write(6,*)

	! NO leading tab appears on this line ! . 'No leading tab above, no syntax error' <tab>write(6,*) <tab>! Leading tab appears on this line! . 'Leading tab above causes syntax error' <tab>stop <tab>end $ ifc –c tab.f program tab ^ Warning 4 at (1:tab.f) : Tab characters are an extension to standard Fortran 95 . 'Leading tab above causes syntax error' ^ Error 24 at (8:tab.f) : syntax error 1 Error compilation aborted for tab.f (code 1)
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034 or higher. You may download and install the latest product update from the Premier Support web site at . You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Calls to FSTAT do not compile as documented
Reference #	33731
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0, 7.1
Operating System	SGI – Linux* for Altix*
Problem Description	The source statement below referencing the instrinsic function fstat, as documented, does not compile. istat = fstat(1, statarray) Compilation generates the following error message: "Error 259 at (4:example.f90) : This function has the wrong number of arguments or arguments with the wrong name, type or rank" The source statement will compile without error if you do not include a "use

	iflport" statement, however, the result returned from fstat may be incorrect.
Resolution/Status	This problem has been resolved in the Intel(R) Fortran Compiler 7.1 l_fc_pc_7.1.022 or later. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	cachesize function returns incorrect values
Reference #	43808
Product	Intel(R) Fortran Compiler for Linux*
Version	7.1
Operating System	Red Hat* Advanced Server 2.1
Problem Description	The following test case when run on a 900Mhz Itanium(R) 2 system with 1.5 MB L3 cache returns incorrect results for the L3 cache size. It should return 1.5 MB rather than 3 MB. write(*,*) "cachesize: level 1 =", cachesize(1) write(*,*) "cachesize: level 2 =", cachesize(2) write(*,*) "cachesize: level 3 =", cachesize(3) End ./cachesize cachesize: level 1 = 16 cachesize: level 2 = 256 cachesize: level 3 = 3072
Resolution/Status	This is resolved in an 8.0 update.

Title	Zero length .o file created upon abnormal compiler termination
Reference #	34961
Product	Intel(R) Fortran Compiler for Linux*
Version	7.1
Operating System	Red Hat* 7.1
Problem Description	If the Intel compiler terminates abnormally, a zero−length .o file may be created. This behavior can lead to further application build issues. For example a makefile upon restart may not deem re−compilation of the corresponding source file in the presence of the zero−length .o file.

Resolution/Status	This is a known issue and may be resolved in a future product release.

Title	Array Section Argument Corrupted at Run Time
Reference #	37124
Product	Intel(R) Fortran Compiler for Linux*
Version	7.0
Operating System	Red Hat* 7.1
Problem Description	When an array section is passed as an argument to a routine that takes it using an assumed−shape dummy argument (for example, a(:,:)), then that routine passes that array to another routine that uses an assumed−size dummy argument (for example, a(N,*)), the array values may become corrupted during runtime.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_pc_7.1.037 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/. As a workaround, use a consistent dummy argument between the two routines (either both assumed−size or both assumed−shape).

Title	Internal compiler error when using the −openmp_report2 compiler switch
Reference #	38019
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	Red Hat* Advanced Server 2.1
Problem Description	Using the compiler option −openmp_report2 will give an internal compiler error on any source code.
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.034_p{e|l}038 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/.

Title	PRESENT intrinsic may return incorrect value for use inside an internal subroutine
Reference #	39520
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	SuSE*
Problem Description	The Intel(R) Visual Fortran PRESENT intrinsic may operate incorrectly inside a nested contained subroutine. It may return a value indicating a non−present argument is actually present or vice versa.
Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	Scalar array assignment doesn't work in nested internal subroutine
Reference #	41381
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	Red Hat* 8.0
Problem Description	Array assignment of multidimensional arrays, where the array bounds are passed as arguments, doesn't work properly at optimizations −O1 or above in code like the following: MODULE mod PRIVATE PUBLIC :: foo CONTAINS SUBROUTINE foo(x, a, b) INTEGER, DIMENSION(2), INTENT(IN) :: a, b REAL, INTENT(OUT) :: x(a(1):b(1), a(2):b(2)) x = 1.0 CALL foofoo PRINT *, x CONTAINS SUBROUTINE foofoo x = 2.0 END SUBROUTINE foofoo END SUBROUTINE foo END MODULE mod PROGRAM test

	USE mod, ONLY : foo REAL, DIMENSION(:,:), ALLOCATABLE :: x ALLOCATE(x(1, 2)) CALL foo(x, LBOUND(x), UBOUND(x)) END PROGRAM test When run, this code will give the result: 2.000000 1.000000 which is not correct. Using –O0 or the Intel(R) Fortran Compiler 7.1 gives the correct result: 2.000000 2.000000
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_pc_8.0.039_pe043.1 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/.

Title	INTEL (flexlm) vendor daemon fails to start
Reference #	37717
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	Red Hat* 9.0
Problem Description	On a system with glibc 2.3 such as Red Hat Linux* 9.0 or Enterprise Linux 3.0, the vendor daemon "INTEL" aborts with the following message: Unknown host: hostname
Resolution/Status	Updated license managers are available on the Intel(R) Premier Support web site at https://premier.intel.com. If you don't have an account, you will need to register your compiler at http://intel.com/software/products/registrationcenter/ The license managers are automatically available if you have support for the Intel(R) C++ or Fortran Compilers for Linux*, otherwise you will need to request access via Premier Support to the "Intel SW Dev Tools License Servers" product. Alternatively, you can use a different host server for your license manager that runs a supported operating system that is not a flavor of Linux with glibc 2.3. Please see http://support.intel.com/support/performancetools/fortran/license.htm for other operating systems that are supported.

Title	Integer calculations of the form 2**k when k is a negative number produce incorrect results
Reference #	41760
Product	Intel(R) Fortran Compiler for Linux*
Version	7.1
Operating System	Red Hat* 8.0
Problem Description	Integer calculations involving the (**) operator produce incorrect result when the power operand is a negative integer of the default storage type (e.g. –4) as shown in the following example: !!! Example – test.f90 PROGRAM exponent INTEGER :: k, k_pow k_pow = 1 k = –1 k_pow = 2**k WRITE(*,*) "k_pow (where k_pow=2^k) = ",k_pow WRITE(*,'("[Floating] 2^–1 = ",F8.2," 2^k (k=–1) = ",F8.2)') 2**(–1),2**k WRITE(*,'("[Integer] 2^–1 = ",I8," 2^k (k=–1) = "I12)') 2**(–1),2**k END !!! <<<< End of example >>> Intel(R) Fortran Compiler for Linux : 7.1.042 1. For storage type = 4(default)*** $ ifc –w –i4 test.f90 –o 7.1.exe program EXPONENT 11 Lines Compiled $ ./7.1.exe k_pow (where k_pow=2^k) = –2147483648 [Floating] 2^–1 = 0.00 2^k (k=–1) = 0.00 [Integer] 2^–1 = 0 2^k (k=–1) = –2147483648 Remark: Output is incorrect
Resolution/Status	This is a known issue with the Intel(R) Fortran Compiler for Linux* version 7.1 with package ID l_fc_pc_7.1.038. A possible workaround is to compile with the –i2 or –i8 compiler options. This issue has been resolved in the Intel(R) Fortran Compiler for Linux* version 8.1 with package ID l_fc_pc_8.1.019 or higher.

Title	fpp flags an error on HOLLERITH constants containing '!'
Reference #	41783
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	Red Hat* Advanced Server 2.1

Problem Description	Example below:
	
	$ cat hollerith.f
	DATA NEXC/1H!/
	end
	$ ifort –O0 –V –fpp –c hollerith.f
	Intel(R) Fortran Itanium(R) Compiler for Itanium(R)–based applications
	Version 8.0 Build 20040212 Package ID: l_fc_pc_8.0.042
	Copyright (C) 1985–2004 Intel Corporation. All rights reserved.
	
	hollerith.f(1): #error: bad hollerith constant.

Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	real*4 arrays with module–variable size used in OpenMP* reductions cause Abort
Reference #	42931
Product	Intel(R) Fortran Compiler for Linux*
Version	8.0
Operating System	SuSE*
Problem Description	Using an OpenMP* reduction clause on an array that has been declared like so: real*4 :: x(n) where n is a module variable will cause the Intel(R) Fortran runtime libraries to issue an "Aborted" message.
Resolution/Status	This is a known issue that may be resolved in a future product release.

Windows*

Title	Unnecessary Float–to–Double Conversions Hinder Performance
Reference#	23927
Product	Intel(R) Fortran Compiler for Windows*

Version	7.0
Operating System	Windows* 2000 Professional
Problem Description	The Fortran Compiler may unnecessarily promote single precision values to doubles using the fma.d instruction prior to using them in subsequent calculations.
Resolution/Status	This problem has been resolved in the Intel(R) Visual Fortran Compiler for Windows* 8.0 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/.

Title	EDB does not recognize executable program path names containing blank characters
Reference#	24678
Product	Intel(R) Fortran Compiler for Windows*
Version	5.0, 6.0, 7.0
Operating System	Windows* 2000 Professional
Problem Description	When invoked via the EDB button on the Microsoft* Visual* C++ 6.0 IDE toolbar, EDB cannot find the executable file to be loaded when any part of the path name contains one or more blank characters. For example the executable file /mydirectory/test cases/test.exe cannot be loaded because in the above path, the directory "test cases" contains a blank character.
Resolution/Status	This is a known issue that may be resolved in a future product release. As a workaround you can load the executable file by using the "Load" option from the "File" menu in EDB.

Title	Character constant with KIND not accepted in format field of WRITE
Reference#	26140
Product	Intel(R) Fortran Compiler for Windows*
Version	6.0,7.0
	Windows* XP Embedded

Operating System	
Problem Description	The compiler doesn't accept a character constant with a kind in the format field of a WRITE statement. A sample test case is shown below: program testcase implicit none write(*,1_'(i2)')3 end
Resolution/Status	This is resolved in the 8.0 compiler.

Title	Apps with source files with names beginning with 'etrip' may fail at runtime
Reference#	29518
Product	Intel(R) Fortran Compiler for Windows*
Version	6.0,7.0
Operating System	Windows* 2000 Professional
Problem Description	Source file names beginning with 'etrip' (for example, etripab.f) can be mis–interpreted by the compiler and lead to unintended uses of the EBP register which may cause unexpected application runtime failures when compiling at optimization level /O1 or above.
Resolution/Status	This problem has been resolved in the Intel(R) Fortran Compiler 7.0 W_FC_PU_7.0.081 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	***Fatal: bad switch value 0 at line XXXX in file proton/libi/f90ioa.c
Reference#	31061
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0
Operating System	Windows* 2000 Professional
Problem Description	Applications with many I/O statements that use run–time formats (i.e. no FORMAT statement) may run into the following error at runtime: "***Fatal: bad switch value 0 at line 1088 in file proton/libi/f90ioa.c".

	The error will occur during an I/O statement in the application.
Resolution/Status	The cause of this issue is an overflow of internal format table, which is used to keep run−time format strings. This problem has been resolved in a product update with package ID W_FC_PC_7.1.025 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/.

Title	QWIN.HLP help file missing from the compiler package
Reference#	31267
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0
Operating System	Windows* XP Embedded
Problem Description	The QWIN.HLP help file is missing from the Intel(R) Fortran Compiler for Windows package. The following test program illustrates this problem. −−−−−−−−−−−−−−−−−− print *,'sample program' end −−−−−−−−−−−−−−−−−− − Build the test program with "ifl /MW test.f" − run test.exe − Select "No" from the "Program Finished" dialog box to keep the windows open. − In the Quick Win application menu, select Help, then Contents.
Resolution/Status	This problem has been resolved in the Intel(R) Visual Fortran Compiler 8.0. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	Unable to compile individual Fortran file in Microsoft* Visual Studio .NET* environment
Reference#	30924
Product	Intel(R) Fortran Compiler for Windows*

Version	7.0
Operating System	Windows* XP Professional
Problem Description	In Microsoft* Visual Studio .NET* it is possible to right click on a C/C++ file and select "compile" to build just that file. There is also a function key for this as well. It is not possible to compile an individual Fortran file, however, with the Intel(R) Fortran Compiler for Windows*.
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_P_8.0.035 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Using BSEARCHQQ Function Causes Syntax Error 394 – Misplaced Statement
Reference#	31493
Product	Intel(R) Fortran Compiler for Windows*
Version	6.0,7.0
Operating System	Windows* XP Embedded
Problem Description	Code using the BSEARCHQQ intrinsic will generate the syntax error 394 for a misplaced statement. This compiler error should not be generated.
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_PC_7.1.025 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com/. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support/.

Title	C preprocessor "stringization" and "merging" operators not recognized by Fortran preprocessor
Reference#	31492
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0
Operating System	Windows* 2000 Professional
Problem Description	The C preprocessor "stringization" operator, "#", and "merging" operator, "##" are not recognized by the Fortran preprocessor. In addition, the comment character, "!", is interpreted as a comment in the preprocessor directive. For example, given the following test program, test.fpp:

	program test
	#define quote(str) #str
	quote(test1)
	#define concat(a,b) a ## b
	concat (1,2)
	#define comment(string) ! string
	comment(test2)
	end
	
	When preprocessing with ifl /E test.fpp the output is:
	
	# 1 "test.fpp"
	program test
	
	#test1
	
	1 ## 2
	
	end
	
	When preprocessing with icl /E test.fpp the (correct) output is:
	
	#line 1 "test.fpp"
	program test
	
	"test1"
	
	12
	
	! test2
	
	end

Resolution/Status	This is a known issue that may be resolved in a future product release. As a workaround in fixed format source for the comment character issue, you can use the "*" or "C" comment characters.

Title	The "build analysis" feature is only available after the initial build of an open solution
Reference#	31556
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0
	Windows* XP Professional

Operating System	
Problem Description	Open a solution in Visual Studio .NET*. On the first build, all Fortran files are rebuilt even if it is not necessary. Subsequent builds do not recompile the Fortran files, but if the solution is closed and reopened all of the Fortran files are rebuilt again.
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_P_8.0.035 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Fortran project relinks on every build
Reference#	31594
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0
Operating System	Windows* XP Professional
Problem Description	Fortran project in Microsoft* Visual Studio* .NET relinks every time a user builds even if everything is up to date.
Resolution/Status	This problem has been resolved in the Intel(R) Fortran Compiler 7.1. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http:/www.intel.com/software/products/support.

Title	EDB and Visual Studio* .NET debugger cannot show local variables in subroutines with entry points
Reference#	32833
Product	Intel(R) Fortran Compiler for Windows*
Version	7.1
Operating System	Windows* XP Professional
Problem Description	When a subroutine contains entry points the local variables are not shown if the entry point is called or an entry point is passed while debugging. This is reproducible with the following console application and by stepping into the SUB1 and SUB2 calls within the debugger. Here the local variables LINE and L cannot be seen either in the local window or the watch window.

	file buf.f: PROGRAM BUG INTEGER I I = 1 CALL SUB2(I) CALL SUB1(I) STOP END file sub1.f: SUBROUTINE SUB1(I) INTEGER I,L CHARACTER*(80) LINE C L = 0 LINE = 'HERE I CAN SE THE LOCAL VARIABLES' L = I GOTO 100 C ENTRY SUB2(I) 100 CONTINUE LINE = 'HERE I CANNOT SE THE LOCAL VARIABLES' L = I RETURN END
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_P_8.0.035 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Compiler interprets \" as escape sequence in command line input
Reference#	33220
Product	Intel(R) Fortran Compiler for Windows*
Version	7.1
Operating System	Windows* XP Professional
Problem Description	An application reads console command line input. The following is representative of the command line input to the application: app.exe –g –dir "C:\" –d 3D The V7.0 and V7.1 Itanium(R) Fortran compilers interpret the following input as:

	arg1: app.exe arg2: –g arg3: –dir arg4: C:" –d 3D Other Fortran compilers interpret the arguments differently: arg1: app.exe arg2: –g arg3: –dir arg4: C: arg5: –d arg6: 3D
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_P_8.0.035 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support. As a workaround add a space after the '\' and before the trailing double quotes to get the intended interpretation with the Intel compiler. For example, app.exe –g –dir "c:\ " –d 3D gives the expected results.

Title	The TRANSFER function produces incorrect result when used in an initialization expression
Reference#	34275
Product	Intel(R) Fortran Compiler for Windows*
Version	7.1
Operating System	Windows* XP Embedded
Problem Description	The TRANSFER function produces incorrect results when used in initialization expressions as shown in the following example: program example character(*),parameter::x(3)=transfer((/32_1,32_1,32_1/),(/' '/)) print *,x print*,len(x)

	print *,'incorrect:',ichar(x) ! Incorrect result print *,'correct:',ichar(transfer((/32_1,32_1,32_1/),(/' '/))) ! Correct result end >ifl example.f90 –out:example.exe example.obj >example 1 incorrect: 0 0 0 correct: 32 32 32
Resolution/Status	This problem has been resolved in a product update with package ID l_fc_p_8.0.032 or higher. You may download and install the latest product update from the premier support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Quad precision complex operands may produce incorrect results when compiled with /Q /QxW
Reference#	34682
Product	Intel(R) Fortran Compiler for Windows*
Version	7.0, 7.1
Operating System	Windows* XP Embedded
Problem Description	Array expressions using quad precision complex operands (COMPLEX*32 in Fortran77 COMPLEX(16) in Fortran 95) sometimes give incorrect results when the code is compile the /QxK or /QxW compiler switches as shown in the following example: >type test.f90 complex(16) a(100),b(100) a=1 b=a+1 print *,sum(b) end >ifl test.f90 >test.exe (200.000000000000000000000000000000,0.000000000000000000000000000000000 Correct result is produced: (200,0) Now, compile the test program with the following commands: On the Intel Pentium® III processor: ifl /QxK test.f90

	On the Pentium® IV processor: ifl /QxW test.f90 >test.exe (0.00000000000000000000000000000000E+0000,200.00000000000000000000000000000000 The result (0,200) will be given. The sum is shifted to the imaginary part.
Resolution/Status	This problem has been resolved in a product with package ID w_fc_p_8.0.035 or higher. You may download and install the latest product update from the premier support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support

Title	Modifying a single file forces rebuild of entire solution
Reference#	36243
Product	Intel(R) Fortran Compiler for Windows*
Version	7.1
Operating System	Windows* 2000 Professional
Problem Description	Using the Intel(R) Fortran Compiler for Windows* 7.1 integrated into the Microsoft* Visual Studio .NET 2003 development environment, modification of a single source file in a project may force a rebuild of all the source files in the solution. The Build option should rebuild only those files that have been modified.
Resolution/Status	This problem has been resolved in a product update with package ID w_fc_p_8.0.035 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support.

Title	Generating debug info for two dimensional array causes internal error
Reference#	38042
Product	Intel(R) Fortran Compiler for Windows*
Version	8.0
Operating System	Windows* XP Advanced Server
Problem Description	The Intel(R) Visual Fortran Compiler for Itanium(R) processor–based applications reports the following error when compiling and generating debug information for code containing two dimensional arrays:

	ifort –Od –Zi –c test.f90 fortcom: Fatal: There has been an internal compiler error (80000002). compilation aborted for spline_fitting_ex1.tsf.f90 (code 1)
Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	16–byte alignment and type declaration causes an internal compiler error
Reference#	39033
Product	Intel(R) Fortran Compiler for Windows*
Version	8.0
Operating System	Windows* XP Professional
Problem Description	Type declarations compiled with the –align:rec16byte switch cause an internal compiler error: module mod_scaling type scale_params real(8) ivstar end type scale_params contains subroutine initscale(gsp) type(scale_params) :: gsp end subroutine end module mod_scaling ifort /c /align:rec16byte test.f90 test.f90(9) : Severe: **Internal compiler error: internal abort** Please report this error along with the circumstances in which it occurred in a Software Problem Report. Note: File and line given may not be explicit cause of this error. subroutine initscale(gsp) _____^ compilation aborted for test.f90 (code 3)
Resolution/Status	This is a known issue that may be resolved in a future product release.

Title	Error when initializing an array using mixed−mode expression
Reference#	41316
Product	Intel(R) Fortran Compiler for Windows*
Version	8.0
Operating System	Windows NT* 4.0 Service Pack 6
Problem Description	Attempts to initialize a REAL array PARAMETER with a mixed−mode array expression may cause the compiler to emit an error: "this operator is invalid in a constant expression evaluation".
Resolution/Status	This problem has been resolved in a product update with package ID W_FC_PC_8.0.044 or higher. You may download and install the latest product update from the Premier Support web site at https://premier.intel.com. You need to be a registered user to access Premier Support. For registration information, please visit http://www.intel.com/software/products/support. As a workaround, eliminate the mixed−mode arithmetic.

Title	GETARG routine does not work correctly with integer*2 data type for its third argument
Reference#	42004
Product	Intel(R) Fortran Compiler for Windows*
Version	8.0
Operating System	Windows* XP Professional
Problem Description	The GETARG routine returns an incorrect value in its third argument if that argument is of type integer*2.
Resolution/Status	This is a known issue that may be resolved in a future product release. The GETARG routine works correctly if you use an integer*4 data type for its third argument.

Title	No warning message issued when division by zero occurs
Reference#	47792
Product	Intel(R) Fortran Compiler for Windows*
Version	8.1
Operating System	Windows NT* 4.0 Service Pack 6
Problem Description	Compiler does not issue a warning message if a division by zero occurs in a constant expression as in the test case below.

	ifort test.f90 >test.exe infinity test.f90 –––––––––– program test real,parameter::one=1,zero=0,bad=one/zero print *,bad end
Resolution/Status	This is a known issue which may be resolved in a future product release